BERKELEY LAB
Bringing Science Solutions to the World

U.S. DEPARTMENT OF
ENERGY
Office of Science

ISOTDAQ 2023 Summary

Luc Le Pottier
June 30, 2023

Highlights!

- Attendance seemed to be approx. **60% grad students**, **35% postdocs**, and a small fraction of undergraduates
- Incredible city for food and tourism, and.. *cats*

In this talk

Today I’ll provide an overview of the latter half of the ISOTDAQ material, which includes 8 labs and 12 lectures. I’ve grouped them roughly into the following sections:

FPGAs

Basic programming & deployment, incl. SoC FPGAs

Data Transfer/Storage

Data transfer protocols and storage technology

Programming

GPUs, Microcontrollers, LabView, & general best practices

System Overviews

Overview of existing TDAQ systems, including space experiments, LHC, and medical imaging

FPGAs

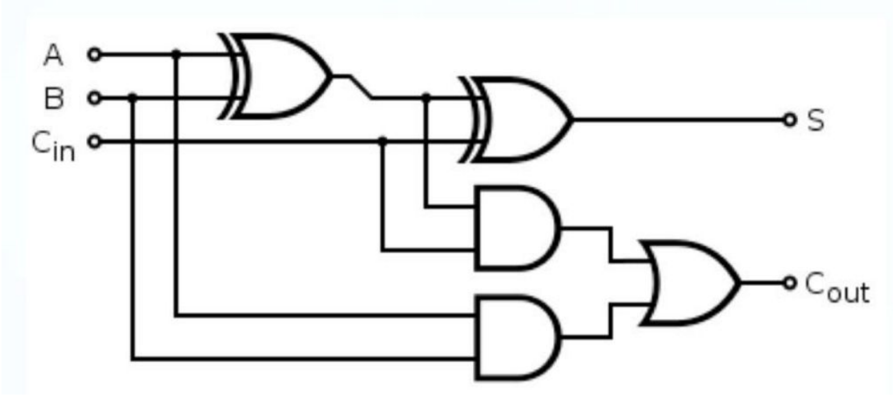
Contributing Tutors:

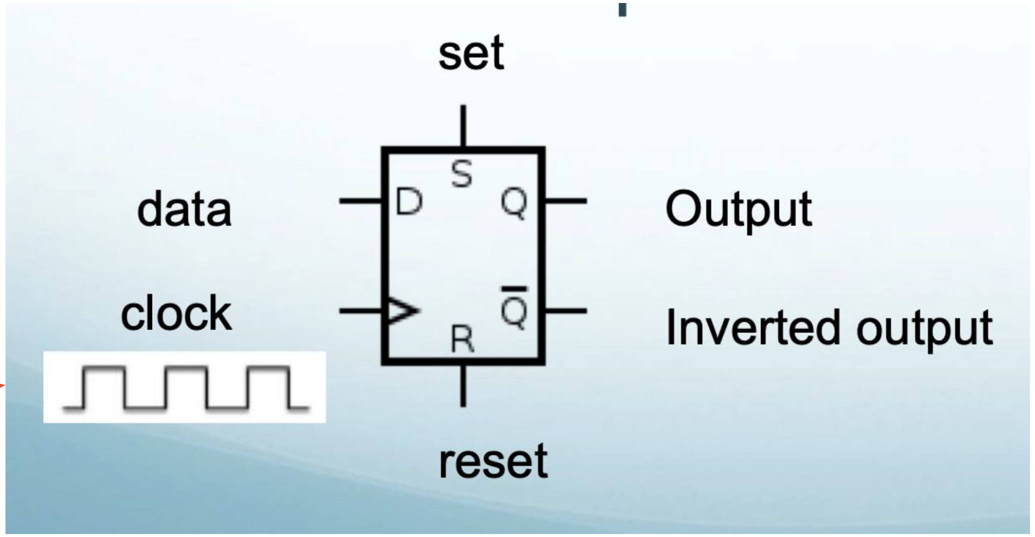
- Hannes Sakulin (CERN)
- Mauricio Feo (CERN)
- Johannes Wuthrich (ETH Zurich)

Overview

(Hannes Sakulin and Mauricio Feo)

- FPGA is essentially a matrix of electrical elements, with configurable wires (interconnects) connecting elements according to **firmware**
- FPGA elements are linked to a **clock**, which facilitates the flow of data through the **intrinsically parallel** system
- Elements include:
  - Look up tables (LUTs), implementing combinational logic with low memory
  - Flip-flops, implementing sequential logic and keeping state (output samples "data" on rising edge, and maintains until next rising edge)
  - Clock tree, distributing timing signal to all elements

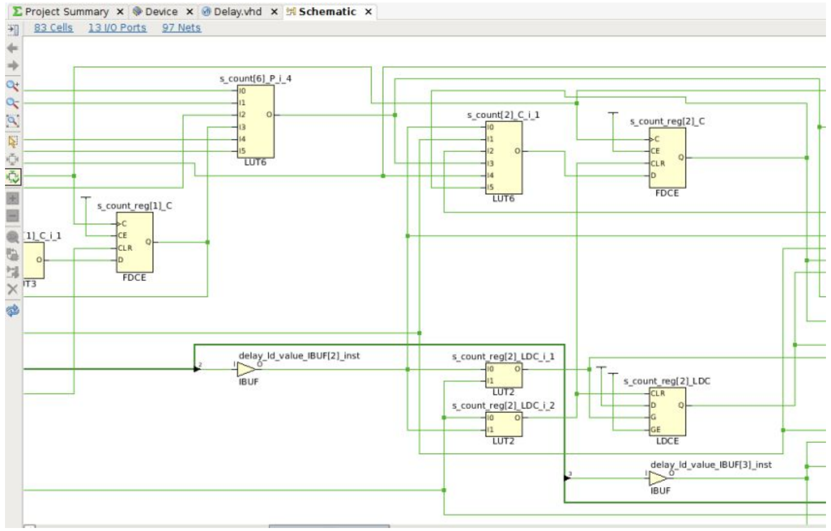




- Input/output pins (programmable)

FPGAs can have millions of logic blocks!

Development Procedure (from labs)



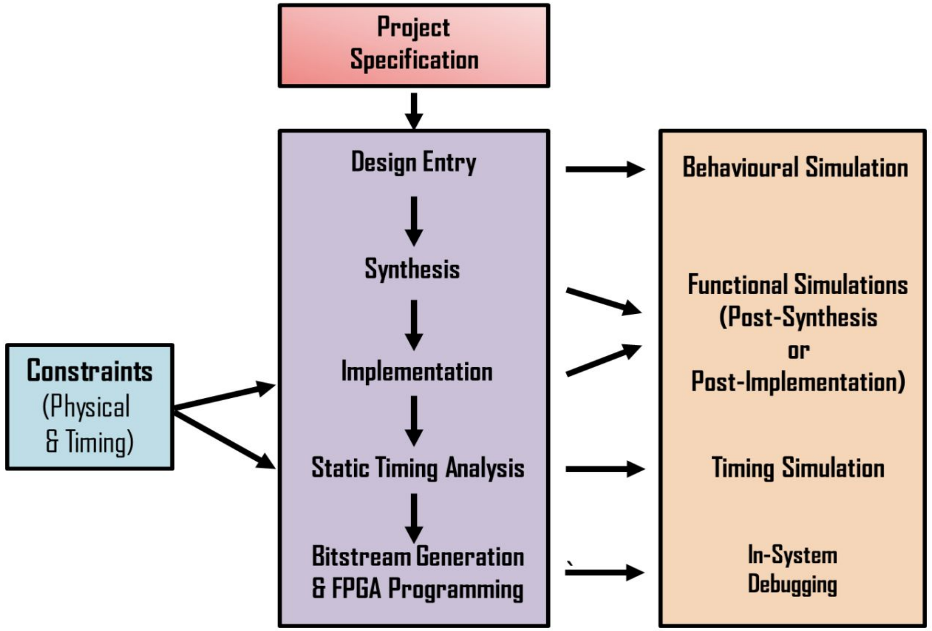
1. Determine project specs, inputs/outputs pin assignment
2. Figure out **hardware/gateware** you want to use.
   – For us, Vivado/QUARTUS in various labs, with ZYNQ/CYCLONE FPGA
3. Construct a block design within the GUI, usually drawing connections by hand
4. Create HDL wrapper for design
5. Edit code further if necessary, then run synthesis (bitstream generation)
6. Run simulation (we used ModelSim) and verify your design, along with timing analysis
7. Send design to FPGA and evaluate results

In general the labs emphasized the **parallel execution** of FPGA gateware, as well as **online configuration** (i.e. SOC FPGA can be easily controlled by on-board CPU running python)

Labs also explored **pitfalls of writing VHDL** (parallel is confusing) and physics examples (tracking)

Usage

- FPGAs are used all over LHC experiments
  - L1 Trigger
  - ML inference
  - Data processing



Project Specification

Design Entry

Synthesis

Implementation

Static Timing Analysis

Bitstream Generation & FPGA Programming

Constraints (Physical & Timing)

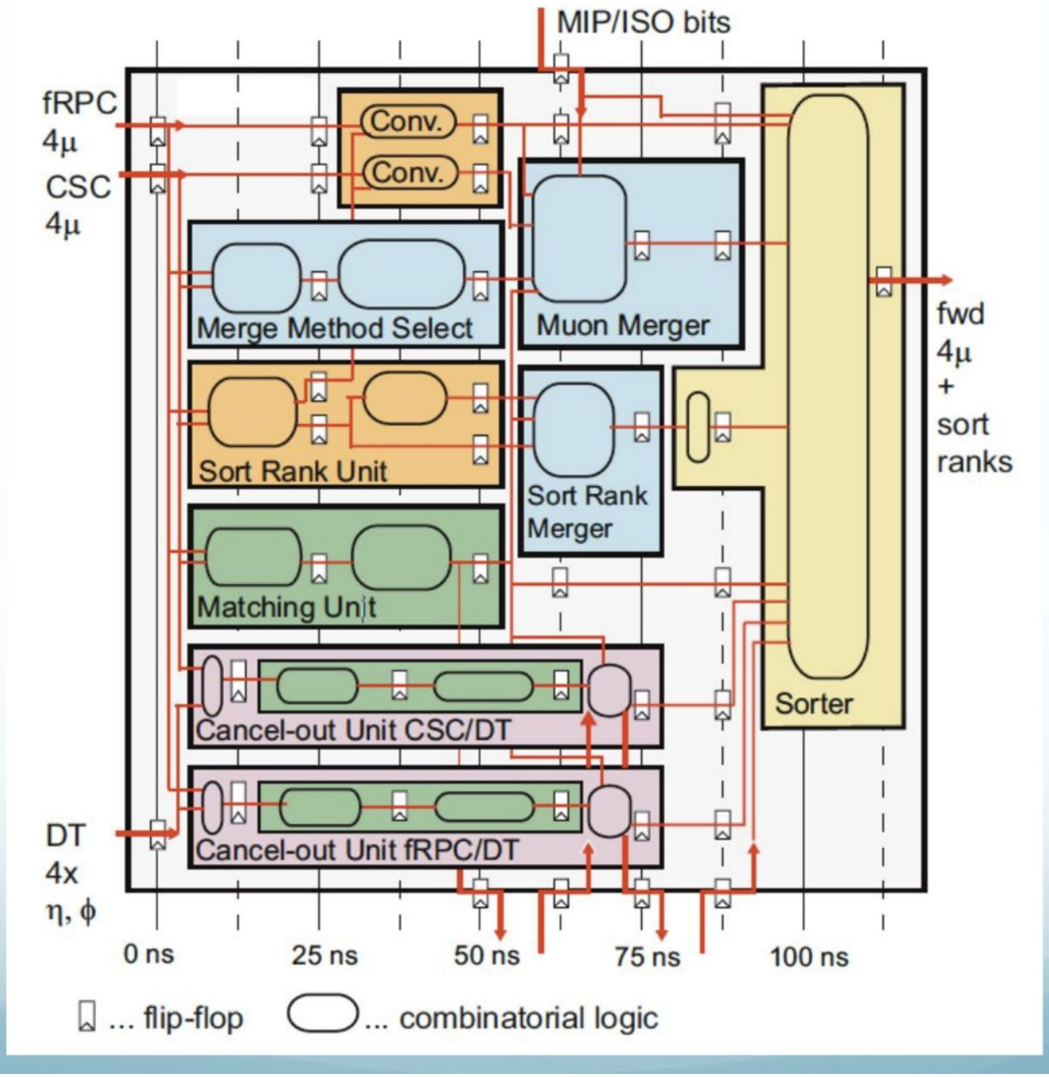
Behavioural Simulation

Functional Simulations (Post-Synthesis or Post-Implementation)

Timing Simulation

In-System Debugging

CMS Global Muon Trigger main FPGA



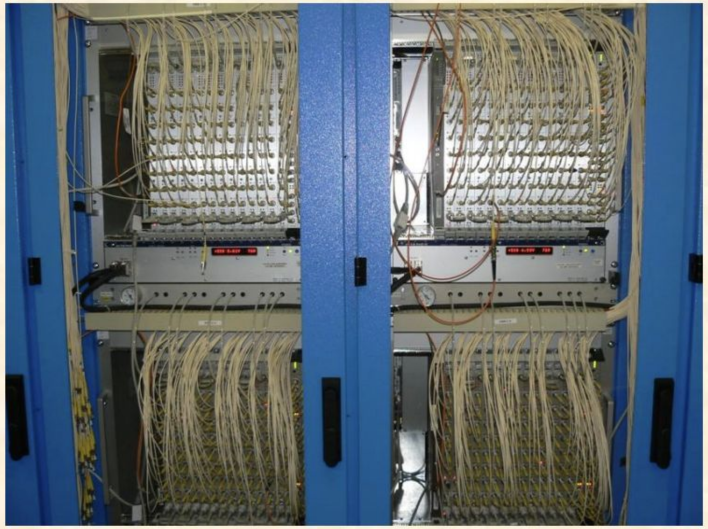
Data Transfer/Storage

Contributing Tutors:

- Paolo Durante (CERN)
- Markus Joos (CERN)
- Adam Abed Abud (CERN)

VMEbus (Markus Joos)

- Lecture with directly associated lab (these usually were best, in my opinion)
- VMEbus is…
  - Easy-to-implement **data transfer** protocol
  - A way to house, power, configure, and readout DAQ electronics systems
- Currently used in >1000 systems at CERN
- Important to know as a **legacy** system, but probably should use MTCA or PCs if you are creating a new experiment
- **Lab**:
  - map VMEbus memory into virtual address space, tying together CPU, PCI, and VMEbus
  - Perform block transfers using a DMA controller (general tech)

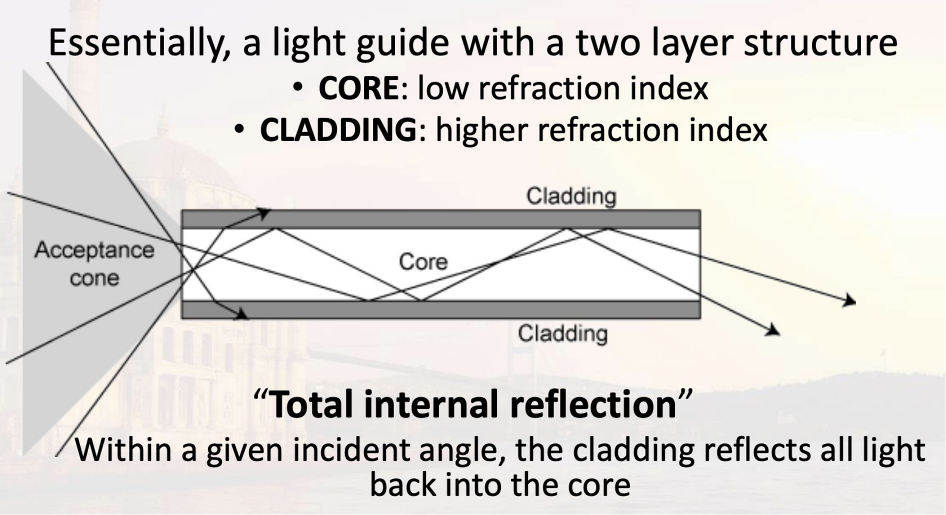


PCIe (Paolo Durante)

- Quite popular still, and being actively updated
  - see e.g. CXL, which provides an interconnect for heterogeneous computing
  - CXL 3.0 in progress, replacing current-gen 2.0 from 2020
- Has “won the interconnect wars”
  - Extremely robust
  - Backwards compatible for decades
  - Connects readout, storage, network, and accelerators (GPU/FPGA)
- In the LHC:
  - ATLAS FELIX cards (also DUNE)
  - LHCb, CMS, etc.
- **Useful to understand if you are working on data transfer in LHC**

Optical Links (Paolo Durante)

- Becoming very popular transfer method
- General properties
  - Cheap material
  - High transfer rate (Tb/s)
  - Low weight/material
  - Long range
  - No interference
  - Actively improving tech
  - Fragile, complex
  - Expensive tech
- LHC usage:
  - lpGBT chipset/protocol
  - Versitale link

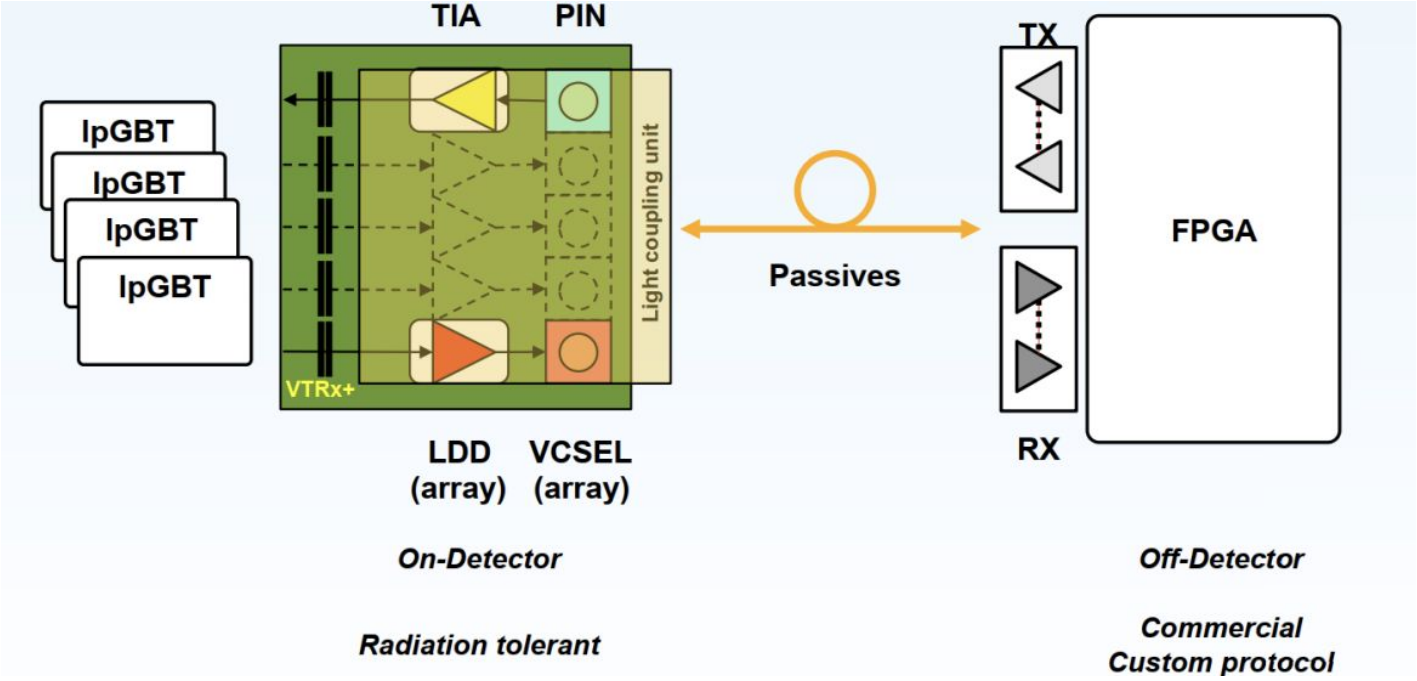


Essentially, a light guide with a two layer structure

- **CORE**: low refraction index
- **CLADDING**: higher refraction index

"Total internal reflection"

Within a given incident angle, the cladding reflects all light back into the core

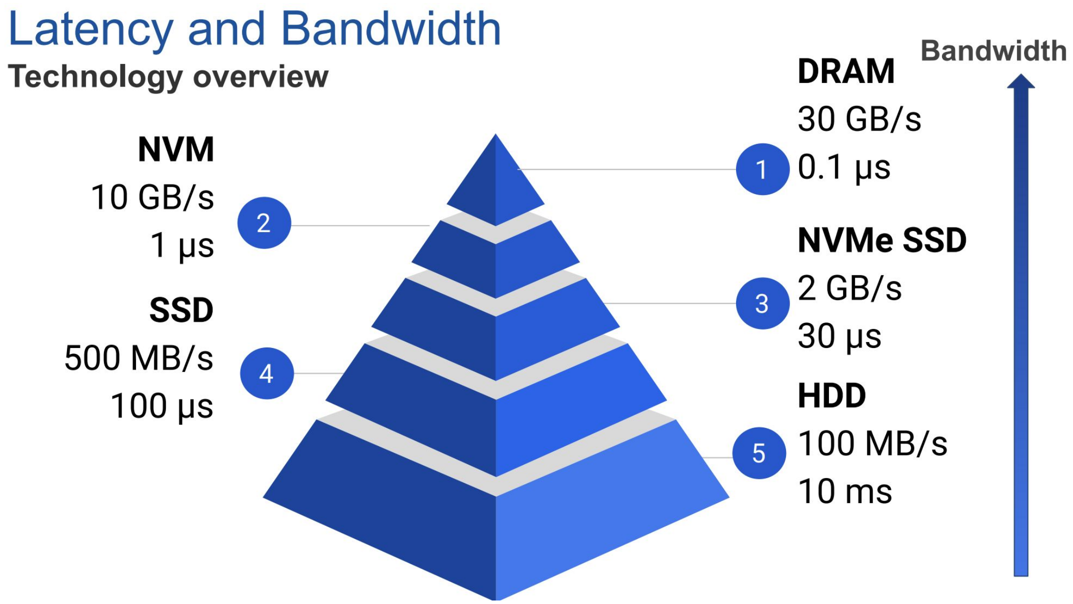


Storage

(Adam Abed Abud)

- For DAQ:
  - Storage systems **enable** physics measurements
  - Choose tech based on total expected data
  - **Throughput**, **latency**, and **cost** also extremely important constraints

- **DUNE storage:**
  - Includes a “supernova buffer,” which is able to readout continuously for O(100)s
  - 1.5Tb per cryostat @10Gb/s throughput
  - Only needed upon supernova triggers (rare)
- **Lab:**
  - Storage rate benchmarking
  - RAID array creation/manipulation
  - Distributed storage systems



Latency and Bandwidth

Technology overview

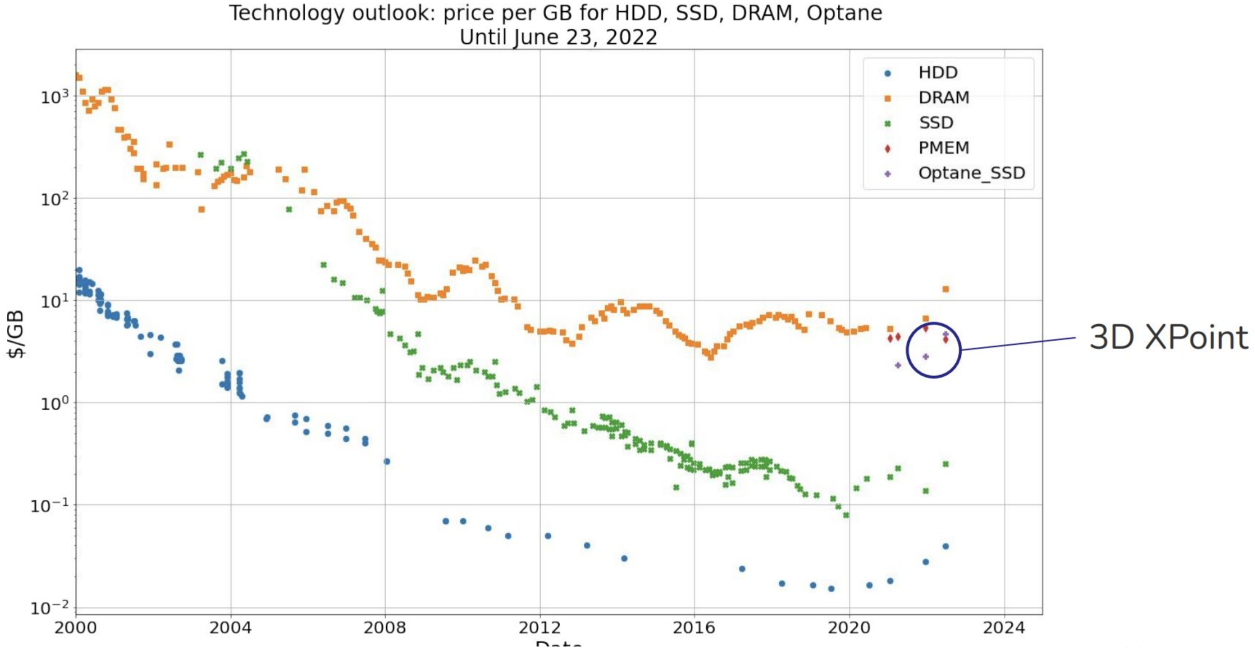
1 **DRAM** 30 GB/s 0.1 µs

2 **NVM** 10 GB/s 1 µs

3 **NVMe SSD** 2 GB/s 30 µs

4 **SSD** 500 MB/s 100 µs

5 **HDD** 100 MB/s 10 ms



Technology outlook: price per GB for HDD, SSD, DRAM, Optane
Until June 23, 2022

3D XPoint

Programming

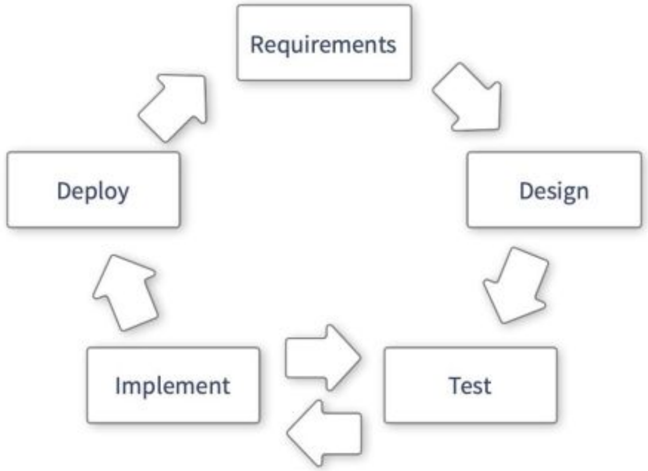
Contributing Tutors:

- Maurício Féo (CERN)
- Gary Boorman (RHU London)
- Dinyar Rabady (CERN)

General Programming (Dinyar Rabady)

- Provided mostly general guidelines for physicists trying to program
  - Avoid duplication
  - Avoid feature bloating
  - Actually **use tests**
  - **Communicate** with users, peers
  - Avoid programming if you can (somebody else has done it)
- Design principles
  - State (and return to often) your project **goals** (are you building too much?)
  - Maintainability (will this break in a year)
  - Scalability (can I use large data volumes – avoid complex data structures)
  - Reusability (possibly creating a library out of part of your code)
- Most helpful thing for me: **Design a test every time you find a bug**
  - Somehow this idea was shocking to me

Test-Driven Development



Requirements

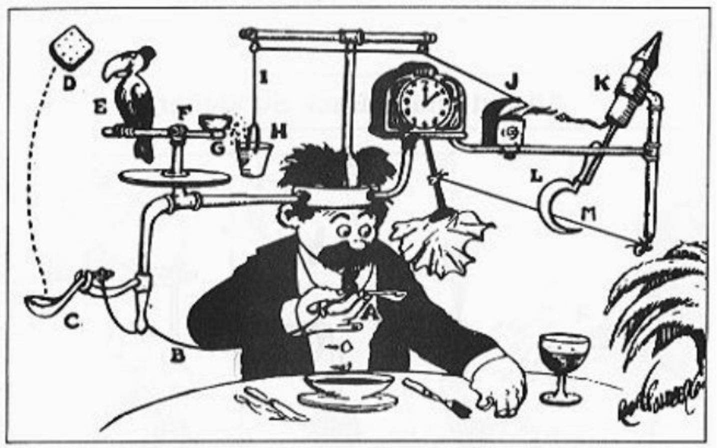
Design

Test

Implement

Deploy

Self-Operating Napkin



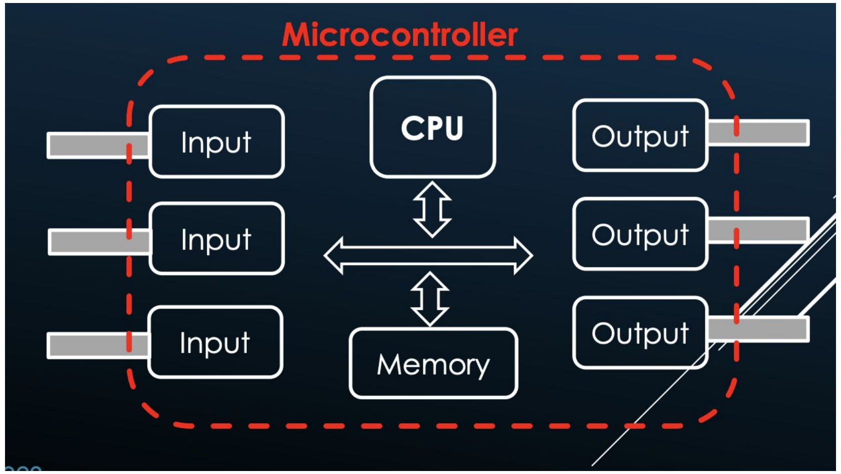
LabView Programming (Gary Boorman)

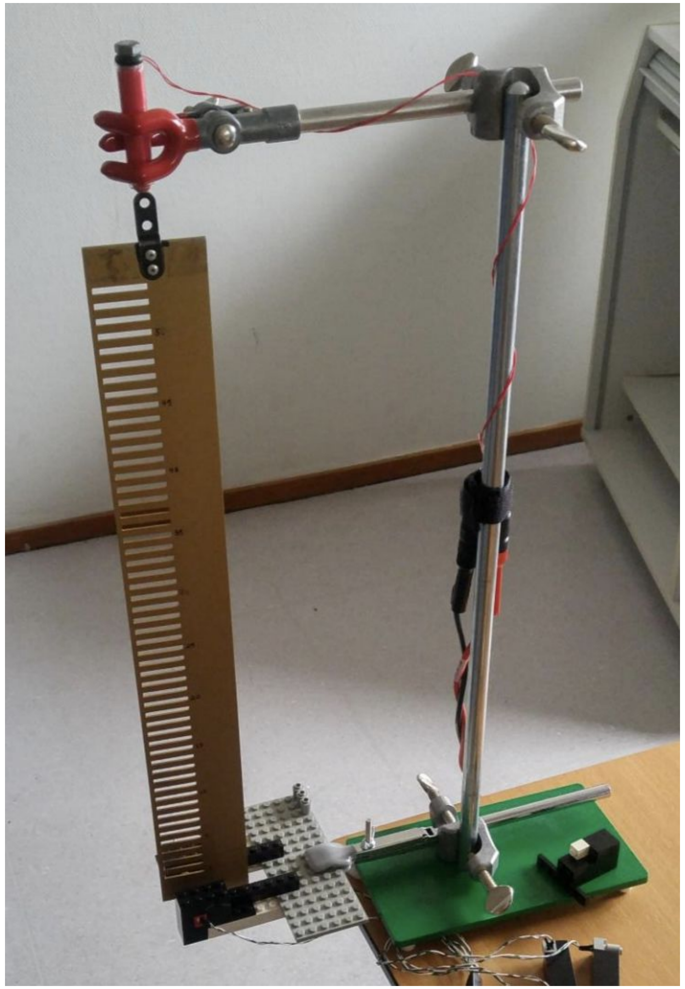
- **LabView** is a graphical programming environment, useful for quick experiment setup/monitoring
- **Biggest positive**: effortless hardware and software integration, with an enormous instrument zoo
- **Biggest negative:** custom hardware is useful
- For FPGA:
  - Labview can be used to program FPGAs
  - Generated HDL code can be user-modified/augmented
  - Used by ICARUS detector at Fermilab, with three FPGAs acquiring at **40 MS/s**
- Lab:
  - Create a LabView experiment/UI
  - Use it to control a small temperature experiment, and send an alarm when certain critical conditions were met

Microcontrollers

(Maurício Féo)

- **Tiny computer** on a single chip
  - CPU, memory, I/O interface, etc.
- Useful because…
  - Cheap, O($1) or less
  - Low power consumption
  - Standalone devices (power only)
  - Can be extremely reliable
- Useful for…
  - Monitoring
  - Control
  - Slow DAQ





- Usually includes ADCs, output signal generators, clock, sleep modes, etc.
- At LBL, we use these for temp/humidity monitoring
- **Lab:**
  - Using Arduino IDE, program a light sensor controller to take data
  - Use this data (along with microcontroller clock) to measure gravity

System Overviews

Contributing Tutors:

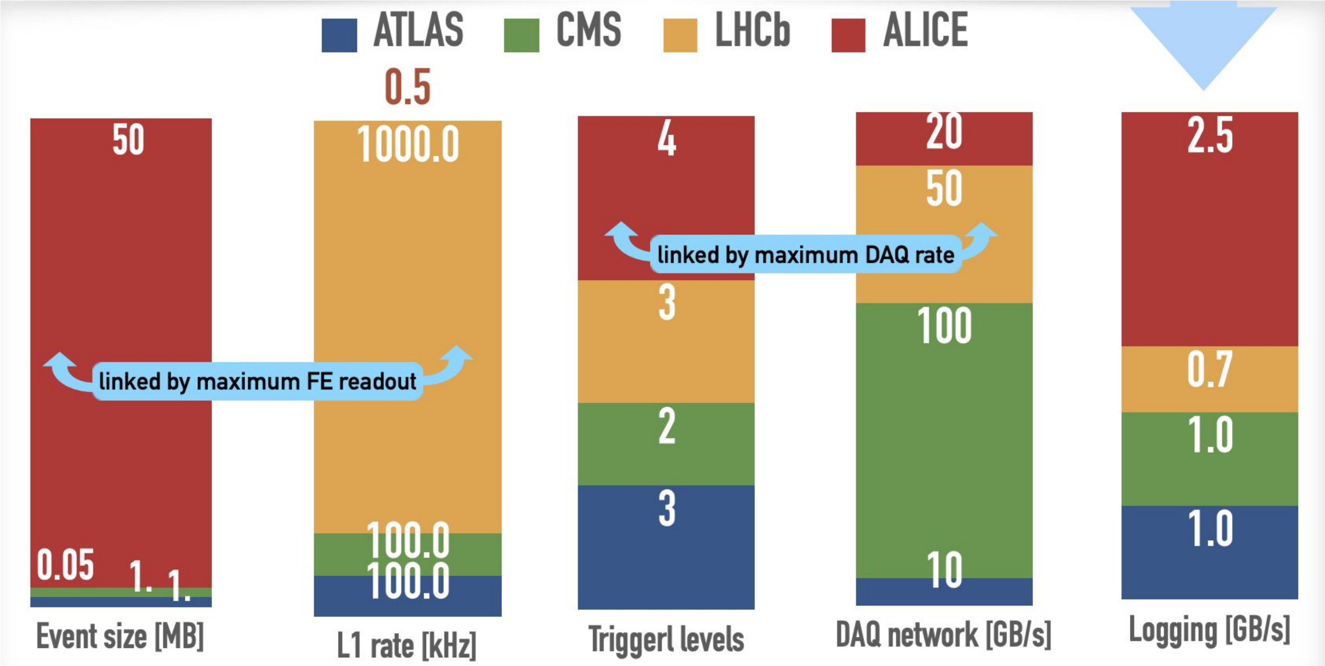
- Valentina Scotti (INFN)
- Francesca Pastore (RHU London)
- Martin Purschke (Brookhaven)

Future of LHC DAQ (Francesa Pastore)

- Luminosity increasing… how to handle?
  - Reduce the amount of data (trigger)
  - Have faster transmission and processing

➡ Robustness and redundancy
➡ Scalability to adapt to Luminosity, detectors,…
➡ Flexibility (10-years experiments)
➡ Based on commercial products
➡ Limited cost

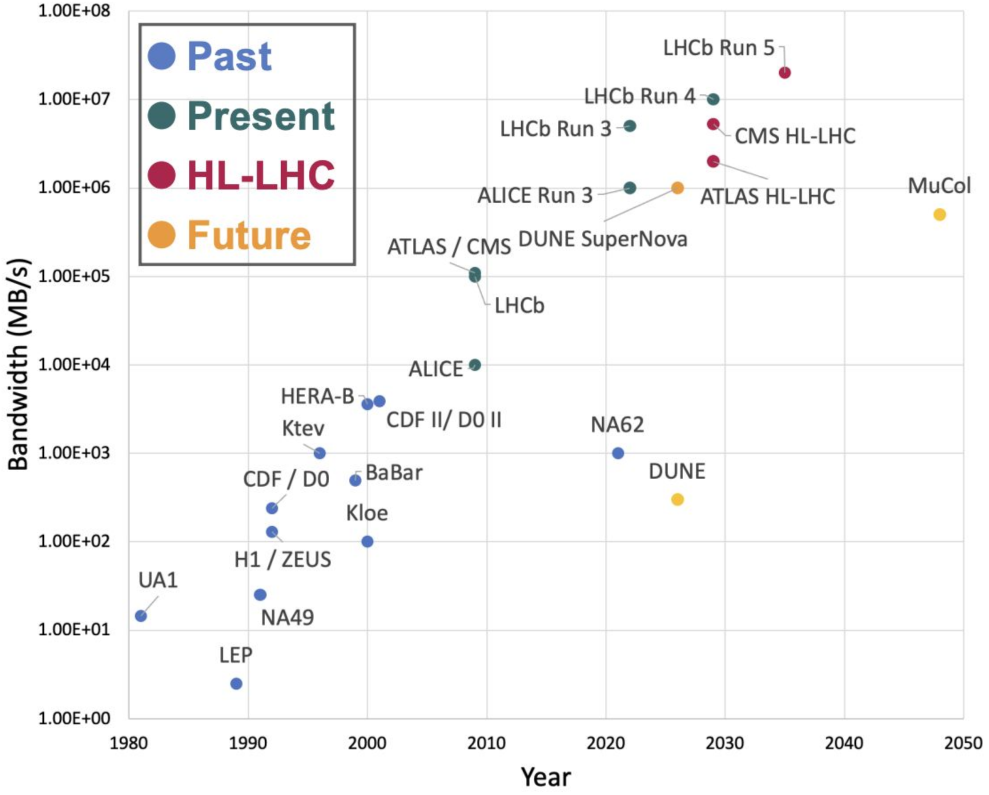
➡ Synchronous: pipeline processing (at fixed latency)
➡ Low latency (fast processing and high speed links)
➡ Scalable
➡ Massively parallel
➡ Bunch Crossing identification capability

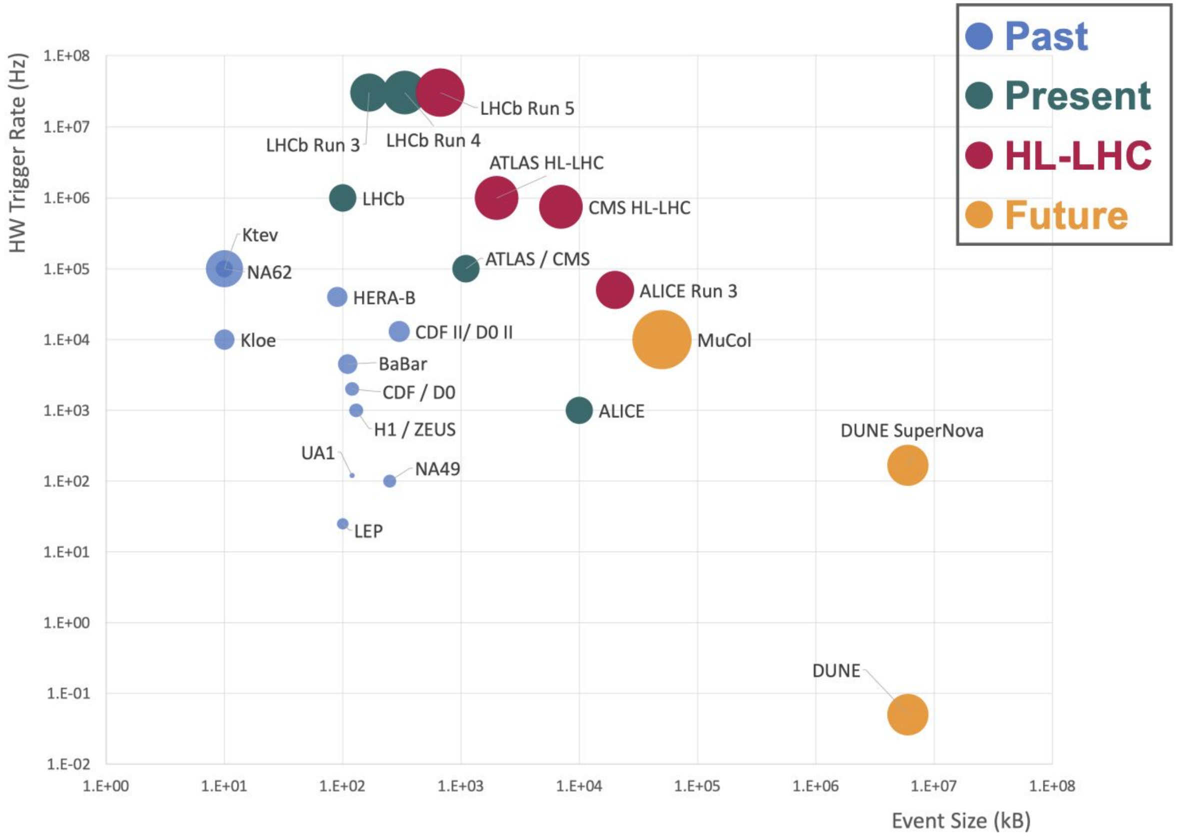


- LHC experiments share a CERN budget for computing resources, constraining trigger/DAQ power together

Future of LHC DAQ: where we are at

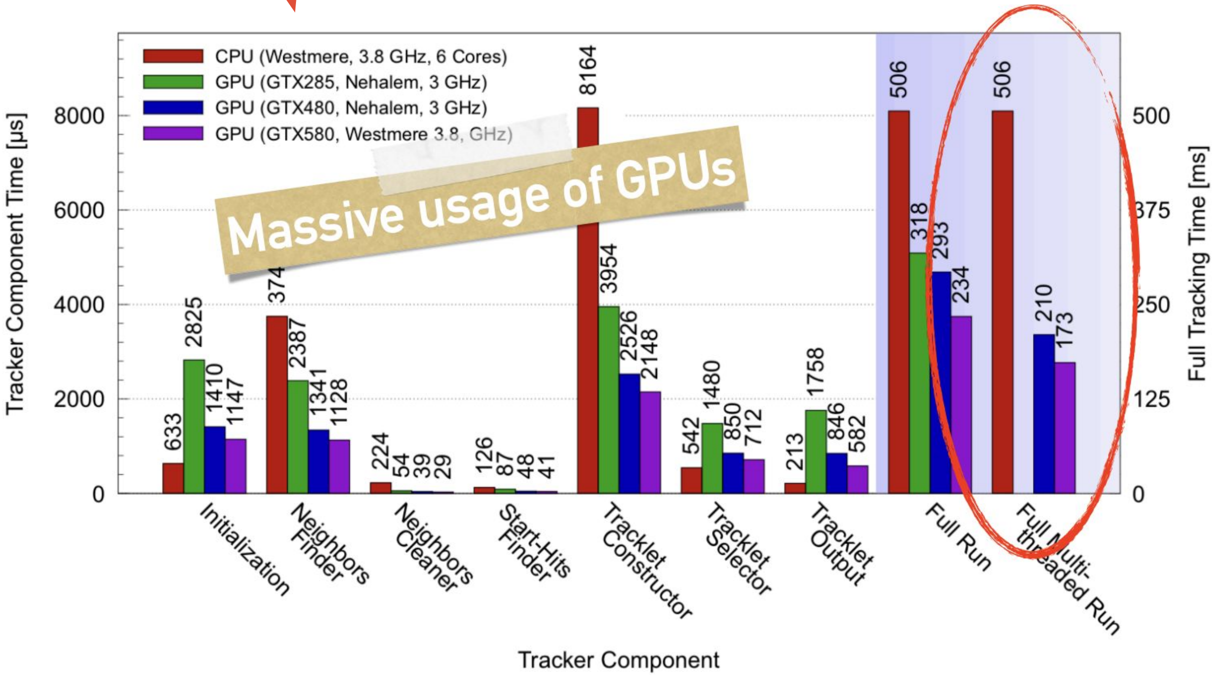
- Need to develop new architectures for increasing data **volumes** + **rates**



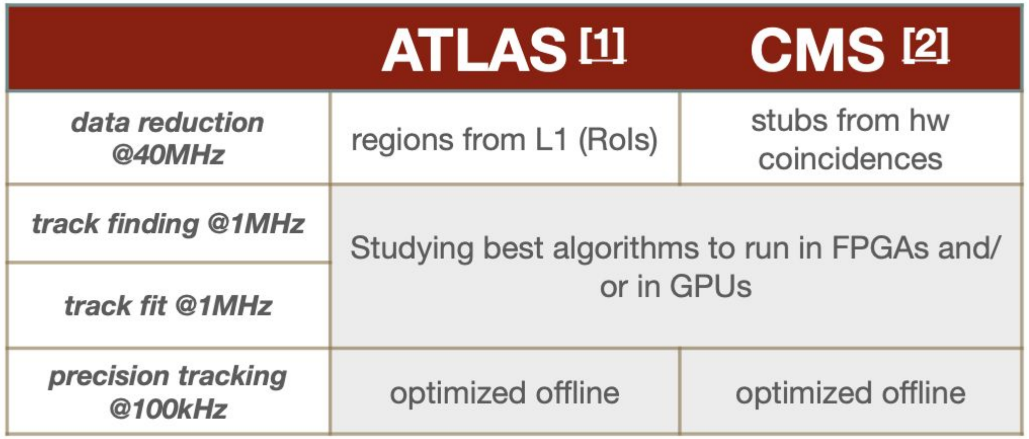


Future of LHC DAQ: Approaches

- **ATLAS/CMS:** high rate readout + event building, robust **track-triggering** systems
- **LHBb**: online/offline merging, huge data throughput
- **ALICE**: **GPUs** for data compression



Massive usage of GPUs



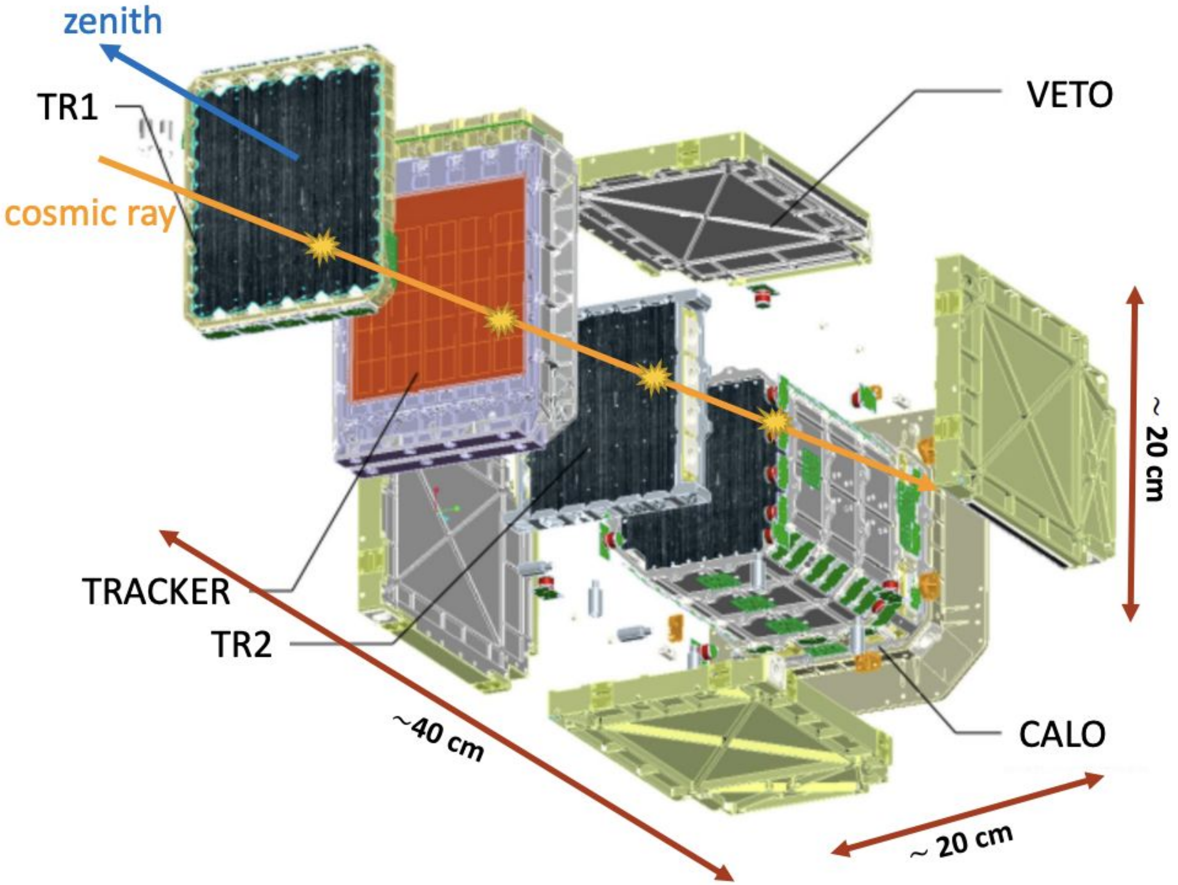
	ATLAS [1]	CMS [2]
data reduction @40MHz	regions from L1 (RoIs)	stubs from hw coincidences
track finding @1MHz	Studying best algorithms to run in FPGAs and/ or in GPUs	
track fit @1MHz		
precision tracking @100kHz	optimized offline	optimized offline

In general: trends towards high bandwidths and commodity hardware, as well as joint efforts

TDAQ for Space

(Valentina Scotti)

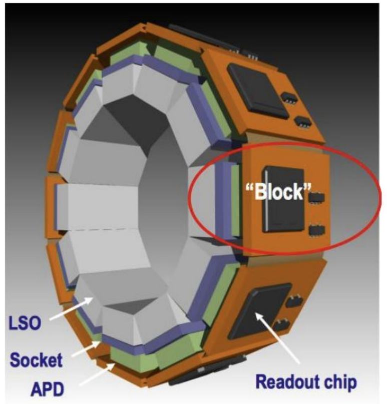
- Focused on **cosmic ray detection**
- Challenges are driven by
  - Reliability issues
  - Weight
  - Power
  - Data transmission BW
- In general, trigger comes from one sub-detector and triggers readout system of all others
- Detector rates are O(1kHz), bandwidth limited

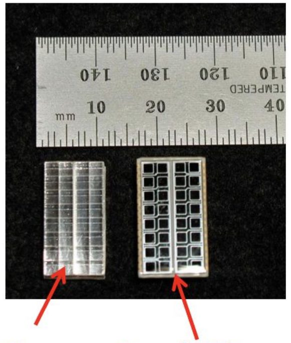


HPED low energy cosmic ray detector

Medical Imaging

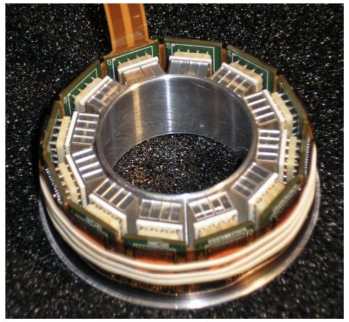
- Positron Emission Tomography (PET) produces back-to photons from ep-annihilation, @511KeV
- O(100ps) timing resolution enough for them, and SiPMs are dominant photomultipliers
- Detector response is main challenge; they do this with **simulation**
- i.e. find P(detector signal | decay position), and invert to find **activity distribution**
- Apparently this is very hard (~30 trillion element matrix)





4x8 array of LSO crystals (2x2x5 mm^3)

APD (Hamamastu S8550)



Actual RatCAP Ring

And they use it to image rats?

Summary of the Summary

The lectures could sometimes be overly focused on hardware specs, rather than takeaways. Overall, though, the vast majority were excellent overviews of their respective subject matter.

The labs were extremely useful. In addition to giving concrete experience implementing many of the lecture topics, the lab manual is a very solid reference for many routine technical procedures (i.e. network configuration, disk config)

And of course, Istanbul was fantastic!

Questions?